

# Bit-Stream Adders and Multipliers for Tri-Level Sigma-Delta Modulators

Chiu-Wa Ng, Ngai Wong, *Member, IEEE*, and Tung-Sang Ng, *Fellow, IEEE*

**Abstract**—We propose both adder and multiplier circuits for bit-stream signal processing customized for tri-level sigma-delta modulated signals. These architectures are the 2-bit extensions from the existing 1-bit bit-stream adders and multipliers, and are shown to offer better signal-to-noise performance. Field-programmable gate array implementations then confirm their efficacy.

**Index Terms**—Adder circuit, multiplier circuit, oversampling, tri-level sigma-delta modulation.

## I. INTRODUCTION

SIGMA-DELTA modulation is a popular technique for analog-to-digital (A/D) and digital-to-analog (D/A) conversion. Traditionally, due to its oversampling nature, the A/D sigma-delta modulated single-bit or short bit-length digital outputs are decimated into multi-bit signals at Nyquist rate and then processed by digital signal processors (DSPs). Similarly, interpolators are used to up-sample the Nyquist rate signals before D/A sigma-delta conversion. Classical decimator and interpolator designs can be found in [1] while recent improved designs are presented in [2], [3].

With the introduction and recent advances of bit-stream signal processing (BSSP) circuits [4]–[9], oversampled bit-stream signals from the sigma-delta modulator (SDM) output can be processed directly without intermediate stages of decimators and interpolators, thereby resulting in resource-efficient signal processing. In [4], [5], two fundamental arithmetic circuits, namely, bit-stream adder and multiplier, are presented. Using these circuits, together with the sigma-delta based low-pass filter [9] and up/down counters as building blocks, a quadrature phase shift key (QPSK) demodulator is developed. A 40% reduction in logic gate count against conventional design is reported. In [6], a resource-efficient phase-locked loop (PLL), composed of the bit-stream adder and multiplier, is presented.

Conventional BSSP circuits are, however, targeted for 1-bit, first-order SDMs [4]–[8]. To improve signal-to-noise performance, one possible way is to increase the number of quantizer bits in the SDM. In [10], tri-level coding is proposed to reduce the quantization noise in a SDM. Besides its higher signal-to-noise ratio (SNR), tri-level  $\{-1, 0, +1\}$  signal representation further inherits the multiplierless nature of binary  $\{-1, +1\}$  code in multiplicative operations, thus leading to highly effi-

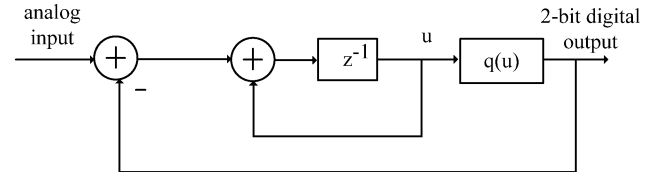


Fig. 1. First-order tri-level SDM.

cient hardware structures. To this end, we investigate the direct processing of tri-level SDM signals. Two novel yet fundamental arithmetic circuits, namely, a bit-stream adder and a multiplier, are proposed in the tri-level context. Field-programmable gate array (FPGA) implementation results on circuit complexity and signal-to-noise-plus-distortion ratio (SNDR) are contrasted against conventional realizations. Finally, application of the proposed adder and multiplier in a digital PLL (DPLL) is demonstrated and contrasted against binary and multi-bit counterparts in terms of circuit complexity.

Fig. 1 shows a first-order tri-level SDM. The input signal is assumed to be bounded in  $[-1, 1]$ , and the quantizer function  $q(u)$  is given by

$$q(u) = \begin{cases} -1, & u < -\alpha \\ 0, & -\alpha \leq u \leq \alpha \\ 1, & u > \alpha. \end{cases}$$

The optimum value of the threshold  $\alpha$  is about 0.2 [10] for A/D conversion and is around 0.25 for filter applications [11]. In [11], it is also shown that the variation of SNR versus  $\alpha$  is relatively small. Consequently, we assume  $\alpha = 0.25$  throughout this brief. To facilitate hardware implementation, 2's complement encoding is used to represent the outputs of the ternary quantizer, namely,  $-1 \rightarrow 11, 0 \rightarrow 00, 1 \rightarrow 01$ .

## II. TRI-LEVEL BIT-STREAM ARITHMETIC CIRCUITS

### A. Tri-Level Bit-Stream Adder

Fig. 2 shows the bit-stream adder proposed in [7] for 1-bit inputs  $A$  and  $B$ . The addition is carried out by a full adder. The circuit outputs the carry bit,  $C_{out}$ , and uses error feedback to reduce the error caused by neglecting the sum bit. Since 2's complement representation is used for the tri-level SDM data stream, the above idea can be directly extended to the 2-bit case. Fig. 3 shows the proposed tri-level bit-stream adder. The 2-bit inputs  $x = [x_1x_0]$  and  $y = [y_1y_0]$  are sign extended to  $[x_1x_1x_0]$  and  $[y_1y_1y_0]$ , respectively, and summed to produce a 3-bit result, which is truncated into the 2-bit output  $z = [z_1z_0]$ . The neglected bit is stored and fed back to the adder to reduce the truncation error.

Manuscript received May 2, 2007; revised June 27, 2007. This work was supported in part by the University Research Committee of The University of Hong Kong. This paper was recommended by Associate Editor G. Manganaro.

The authors are with the Department of Electrical and Electronic Engineering, The University of Hong Kong, Hong Kong (e-mail: cwng@eee.hku.hk; nwong@eee.hku.hk; tsng@eee.hku.hk).

Digital Object Identifier 10.1109/TCSII.2007.906173

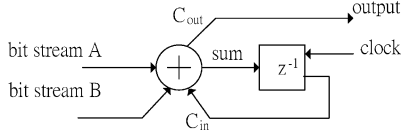


Fig. 2. Bit-stream adder using noise shaping [7].

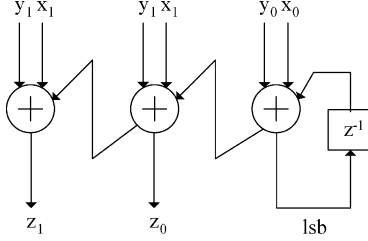


Fig. 3. Proposed tri-level bit-stream adder.

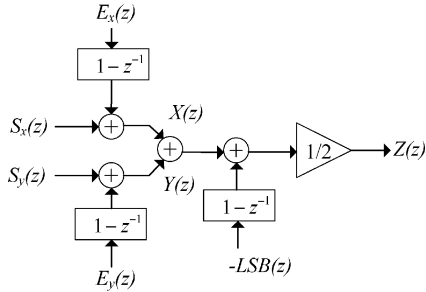


Fig. 4. Equivalent model of the tri-level bit-stream adder.

The same adder can be used to perform bit-stream subtraction, the subtrahend is simply negated. The following logic equations can be shown to perform negation:

$$z_0 = x_0, \quad z_1 = \bar{x}_1 x_0$$

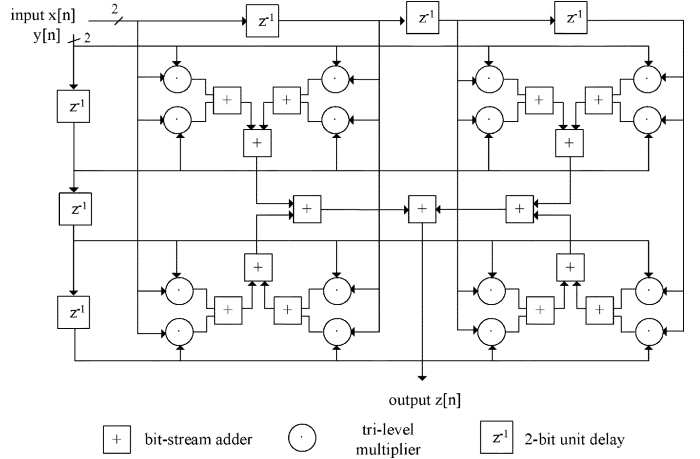
where  $[z_1 z_0]$  denotes the 2-bit output while  $[x_1 x_0]$  denotes the 2-bit inputs. By  $z$ -transform (where capital letters are used to denote the  $z$ -domain variables, thus  $z[n] \xrightarrow{z\text{-transform}} Z(z)$  etc.), similar to the case in [7], the output is given by

$$Z(z) = \frac{X(z) + Y(z) - (1 - z^{-1})\text{LSB}(z)}{2}. \quad (1)$$

An equivalent model of the bit-stream adder is shown in Fig. 4. Both  $X(z)$  and  $Y(z)$  consist of the signal components,  $S_x(z)$  and  $S_y(z)$ , plus the quantization noise components,  $E_x(z)$  and  $E_y(z)$ , that are shaped by the first-order noise transfer function (NTF)  $(1 - z^{-1})$ . The bit-stream adder adds another noise source  $-\text{LSB}(z)$ , also shaped by  $(1 - z^{-1})$ .

Assuming that the quantization noise is white and uniformly distributed in the range  $[-\Delta/2, \Delta/2]$ , where  $\Delta$  is the quantizer step size ( $\Delta = 1$  in this brief), the power-spectral density (PSD) of the input noise is given by [12]

$$P_e(f) = \frac{\Delta^2}{12} [2 \sin(\pi f)]^2 = \frac{1}{3} \sin^2(\pi f). \quad (2)$$

Fig. 5. Tri-level bit-stream multiplier for  $L = 4$ .

Although the truncation noise  $-\text{LSB}$  has only two values, namely,  $-1$  and  $0$ , it is assumed, for simplicity, to be white and uniformly distributed in the range  $[-1, 0]$ . As it is shaped by the function  $(1 - z^{-1})$ , the PSD of the truncation noise is also given by (2). Further assuming that the three noise sources are independent, the total noise power is the summation of them. With a gain of  $1/2$  at the output, the output noise PSD of the bit-stream adder is then estimated as

$$P_o(f) = \frac{1}{3} \sin^2(\pi f) \cdot 3 \cdot \left(\frac{1}{2}\right)^2 = \frac{1}{4} \sin^2(\pi f). \quad (3)$$

### B. Tri-Level Bit-Stream Multiplier

The bit-stream multiplier presented in [4] performs the multiplication of two bit-stream signals,  $x[n]$  and  $y[n]$ , through the following operation:

$$z[n] = \left[ \frac{1}{L} \sum_{i=n-L+1}^n x[i] \right] \left[ \frac{1}{L} \sum_{j=n-L+1}^n y[j] \right] \quad (4)$$

where  $L$  is the length of the time interval. To avoid multi-bit multiplier, (4) is rewritten into the following form:

$$z[n] = \frac{1}{L^2} \sum_{i,j=n-L+1}^n x[i]y[j]. \quad (5)$$

It can be seen from (5) that the multiplier consists of a bit-stream adder tree that computes the summation of the sub-products  $x[i]y[j]$ . With several architectural modifications, the same principle can be applied to the case of tri-level bit-stream multiplication.

The proposed multiplier circuit with  $L = 4$  is shown in Fig. 5 while the original one can be found in [4]. (The choice of an appropriate  $L$  will be discussed in later sections.) Compared with the multiplier in [4], the new multiplier operates on 2-bit input and output signals. Each unit delay involves 2 flip flops (FFs). The proposed tri-level bit-stream adder in Section II-A is used in place of the binary bit-stream adder in the adder tree. Finally,

to produce the sub-product  $x[i]y[j]$ , a tri-level multiplier is designed instead of the exclusive-OR gate in [4]. The tri-level multiplier performs the multiplication of two tri-level  $\{-1, 0, 1\}$  inputs. Using 2's complement representation and treating the code "10" as "don't care," the following logic equations are obtained:

$$z_0 = x_0y_0, \quad z_1 = \bar{x}_1x_0y_0 + x_1\bar{y}_1y_0$$

where  $[z_1z_0]$  denotes the 2-bit output while  $[x_1x_0]$  and  $[y_1y_0]$  denote the 2-bit inputs.

The output noise-spectral density of the bit-stream multiplier can be estimated using the equivalent adder model in Fig. 4. Assuming that the truncation noise  $-\text{LSB}(z)$  in each bit-stream adder is independent, the PSD of the total noise contribution due to the adder tree can be obtained as follows:

$$\begin{aligned} P_a(f) &= P_e(f) \left[ 2^{l-1} \cdot \left( \frac{1}{2^l} \right)^2 + 2^{l-2} \cdot \left( \frac{1}{2^{l-1}} \right)^2 + \cdots + \frac{1}{4} \right] \\ &= P_e(f) \left[ \frac{1}{2} - \frac{1}{2^{l+1}} \right] \end{aligned} \quad (6)$$

where  $l = 2 \log_2 L$  denotes the number of levels in the adder tree. The total noise-spectral density of the bit-stream multiplier can be obtained as in [4]

$$\begin{aligned} E_m(f) &= (H(e^{j2\pi f})S_x(e^{j2\pi f})) * (H(e^{j2\pi f})E_y(e^{j2\pi f})) \\ &\quad + (H(e^{j2\pi f})S_y(e^{j2\pi f})) * (H(e^{j2\pi f})E_x(e^{j2\pi f})) \\ &\quad + (H(e^{j2\pi f})E_x(e^{j2\pi f})) * (H(e^{j2\pi f})E_y(e^{j2\pi f})) \\ &\quad + \sqrt{P_e(f) \left[ \frac{1}{2} - \frac{1}{2^{l+1}} \right]} \end{aligned} \quad (7)$$

where  $H(e^{j2\pi f}) = (1/L) \sum_{m=0}^{L-1} e^{-j2\pi mf}$  and  $*$  denotes the convolution operator. The output noise-spectral density consists of several noise components formed by the convolution of the filtered signals or quantization noises. The noise filtering performance of  $H(e^{j2\pi f})$  can be improved by increasing  $L$ . This reduces the output noise-spectral density and hence improves the SNDR performance of the bit-stream multiplier. However, increasing  $L$  leads to a higher hardware complexity. A tradeoff analysis will be shown in Section IV.

### III. FPGA IMPLEMENTATION RESULTS

The proposed bit-stream adder and multiplier are implemented with the new-generation Xilinx Virtex-5 XC5VLX30, which features 6-input look-up tables (LUTs) [13], using the design tool ISE WebPACK 9.1i. Table I presents the implementation results for the conventional (bi-level) and the proposed (tri-level) designs, with  $L = 4$  in the multipliers. For the tri-level bit-stream adder, it can be seen from Fig. 3 that each of the three 1-bit outputs is a function of up to five 1-bit inputs (viz.,  $x_0, x_1, y_0, y_1$  and the unit-delayed LSB). Therefore, each function can be mapped onto a single LUT. The tri-level bit-stream adder can operate as fast as the bi-level one and only requires one more LUT. For the tri-level bit-stream multiplier, the 2-bit architecture is more complicated and, therefore, results

TABLE I  
IMPLEMENTATION RESULTS ON A XILINX VIRTEX-5 XC5VLX30 FOR LOGIC UTILIZATION OF LOOK-UP NUMBER OF FLIP FLOPS (FFS) AND CLOCK FREQUENCY (MHZ)

	Adder		Multiplier	
	bi-level	tri-level	bi-level	tri-level
no. of LUTs	2	3	31	59
no. of FFS	1	1	21	27
freq. (MHz)	1053	1053	351	278

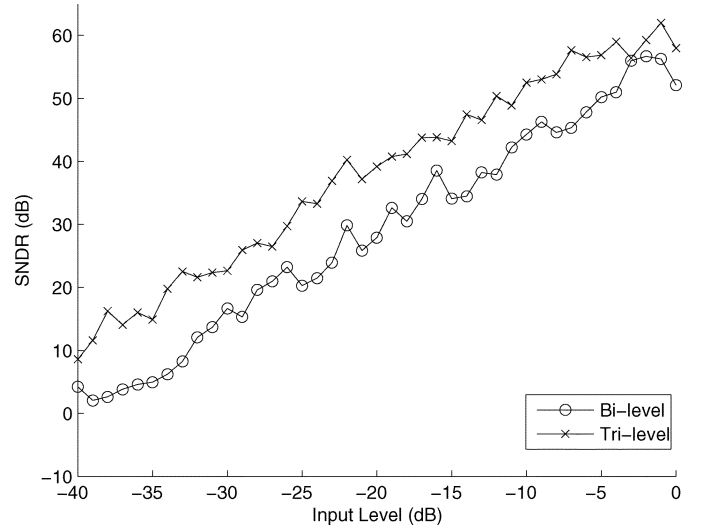


Fig. 6. SNDR versus input amplitude for bi-level and tri-level bit-stream adders.

in increased hardware and lower clock frequency. Such rise in complexity, however, is justified by the significant increase in SNDR, as will be shown below.

### IV. SIMULATION RESULTS

Matlab simulations are carried out to verify the validity of the proposed circuits and compare their performance improvement over conventional designs. Figs. 6 and 7 plot the SNDR against input amplitude curves for the conventional and the proposed bit-stream adders and multipliers, respectively. Throughout the simulations, the two input signals are used which are sinusoidal waves at normalized frequencies of  $3.1 \times 10^{-4}$  and  $2.1 \times 10^{-3}$ , respectively, with equal amplitude. The oversampling ratio (OSR) is 128. The sigma-delta modulated sinusoidal signals with varying input amplitudes are fed into the bit-stream adders and multipliers. A 16384-point fast Fourier transform (FFT) using Hanning window is applied to the resulting bit-stream outputs to obtain the PSDs. The SNDR is calculated as the ratio of the total output signal power over the total in-band noise. Note that the 0 dB signal power level refers to the power when the amplitude of each input sinusoid is unity. Compared with the bi-level counterparts, the proposed tri-level adder and multiplier have average performance gains of 9.0 and 7.8 dB, respectively. These results are consistent with the 7 dB performance gain obtained when ternary SDM is used in the filter design [11]. As increasing the number of bits in the quantizer by one generally leads to an SNDR improvement of more than

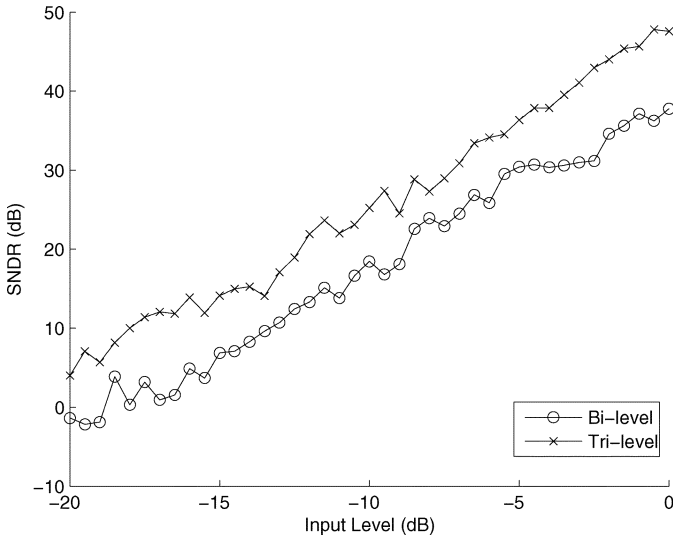


Fig. 7. SNDR versus input amplitude for bi-level and tri-level bit-stream multipliers ( $L = 4$  in both multipliers).

TABLE II  
IMPLEMENTATION RESULTS OF BI-LEVEL ( $L = 8$ )  
AND TRI-LEVEL ( $L = 4$ ) BIT-STREAM MULTIPLIERS

	bi-level	tri-level
no. of LUTs	117	59
no. of FFs	77	27
freq. (MHz)	331	278

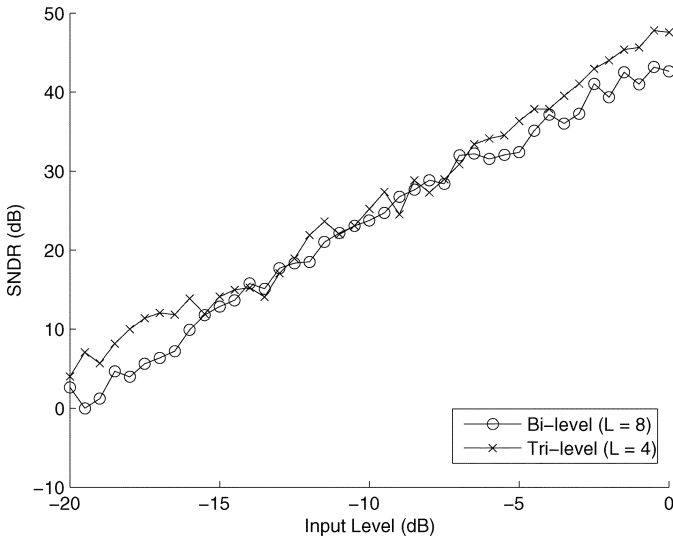


Fig. 8. SNDR versus input amplitude for bi-level ( $L = 8$ ) and tri-level ( $L = 4$ ) bit-stream multipliers.

6 dB, the proposed tri-level BSSP circuits can achieve better performance gain over the bi-level designs.

In Section II-B, it has been noted that the parameter  $L$  affects the SNDR performance and the hardware complexity of the bit-stream multiplier. To evaluate the hardware resource utilization efficiency of the tri-level bit-stream multiplier, Table II contrasts the implementation results for the bi-level bit-stream multiplier

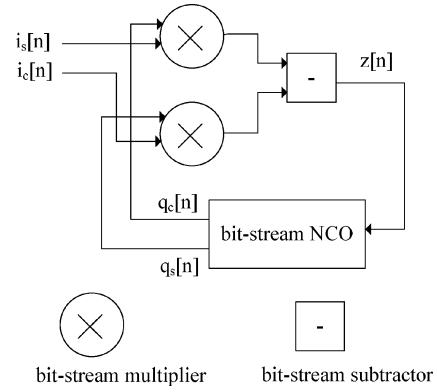


Fig. 9. Type-1 bit-stream DPLL.

for  $L = 8$  and the proposed tri-level design for  $L = 4$ . The SNDR curves against input amplitude for the two multipliers are shown in Fig. 8. While consuming much less resources, the tri-level bit-stream multiplier for  $L = 4$  still achieves better signal-to-noise performance than the bi-level one for  $L = 8$ . Consequently, despite the tri-level design requires higher circuit complexity, it is in fact more efficient than the bi-level design in terms of performance improvement.

## V. APPLICATION EXAMPLE

To demonstrate the application of the proposed bit-stream adder and multiplier, a Type-1 DPLL [14] is implemented. Fig. 9 shows the block diagram of the bit-stream DPLL whose structure is similar to the one presented in [6].

The input signal is assumed to be a complex sinusoid of the form

$$i[n] = i_c[n] + j \cdot i_s[n]. \quad (8)$$

The output of the bit-stream numerically controlled oscillator (NCO) is given by

$$q[n] = q_c[n] + j \cdot q_s[n]. \quad (9)$$

The output phase of the NCO is

$$\theta[n] = \sum_{p=0}^n \frac{1}{K_0 - K_d \cdot z[p]} \quad (10)$$

where  $K_0$  and  $K_d$  are design parameters and

$$z[n] = \text{Im}(i[n] \cdot q[n]^*) \quad (11)$$

where  $\text{Im}(x)$  denotes the imaginary part of  $x$  and  $x^*$  denotes the conjugate of  $x$ . Details of the bit-stream NCO can be found in [4], [6]. By referring to Fig. 4 of [4] or Fig. 6 of [6], the extension of this NCO to a ternary design, specifically, by modifying the digital SDM and up/down counters to their 2-bit (i.e., tri-level) versions, is trivial. In fact, using the proposed bit-stream adder and multiplier as building blocks, other circuit modules

TABLE III  
IMPLEMENTATION RESULTS OF BI-LEVEL, TRI-LEVEL,  
AND 8-BIT DPLL DESIGNS

	bi-level	tri-level	8-bit
no. of LUTs	122	208	351
no. of FFs	79	91	28

in [4]–[6] are readily extended to their ternary counterparts. The phase detector  $z[n]$  is realized by two bit-stream multipliers and a bit-stream subtractor (essentially the same as an adder, see Section II-A). In this particular implementation, the normalized input frequency is  $1/512$ ,  $K_0$  and  $K_d$  are accordingly set to 82 and 5, respectively [4], [6]. The OSR is 128. Simulations confirm that both bi-level and tri-level systems can synchronize to the input signal at steady state. The SNDRs of the bi-level and tri-level DPLL outputs are 35.5 dB and 46.7 dB, respectively. The SNDR is obtained using 16384-point FFT and Hanning window on the output signal  $q_c[n]$  at steady state. Thus, the tri-level design is roughly equivalent to an 8-bit system. To evaluate the merit of using tri-level BSSP technique, an 8-bit DPLL is implemented. Since the NCO in Fig. 9 is only for BSSP implementation, for the 8-bit design, the NCO is implemented as a direct digital synthesizer (DDS). Hardware optimization techniques presented in [15], [16] have been used.

Table III shows the FPGA implementation results of the bi-level, tri-level and 8-bit DPLL designs. It can be seen that both the bi-level and tri-level designs require significantly less LUTs than the multi-bit design at the expense of more FFs. In FPGA implementation, LUT is the resource limiting factor. Moreover, the hardware resources of the necessary decimator and interpolator for the multi-bit system are not counted in this implementation. Therefore, the two bit-stream implementations are much more hardware-efficient than the multi-bit counterpart.

## VI. CONCLUSION

In this brief, a bit-stream adder and a multiplier customized for the tri-level sigma-delta modulated signal processing have been proposed. Hardware architectures have been described and implemented on FPGAs. Compared with the conventional

bi-level designs, the proposed designs have been shown to be more hardware-efficient under the same SNDR requirement. An application example in DPLL has further demonstrated the efficacy and practical interests of tri-level BSSP.

## REFERENCES

- [1] E. Hogenauer, "An economical class of digital filters for decimation and interpolation," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 29, no. ASSP-2, pp. 155–162, Feb. 1981.
- [2] M. Laddomada, "Generalized comb decimation filters for  $\Sigma\Delta$  A/D converters: Analysis and design," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 54, no. 5, pp. 994–1005, May 2007.
- [3] G. Jovanovic-Dolecek and S. K. Mitra, "A new two-stage sharpened comb decimator," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 52, no. 7, pp. 1414–1420, Jul. 2005.
- [4] H. Fujisaka, R. Kurata, M. Sakamoto, and M. Morisue, "Bit-stream signal processing and its application to communication systems," *Proc. IEEE, Circuits, Devices Syst.*, vol. 149, no. 3, pp. 159–166, 2002.
- [5] H. Fujisaka, M. Sakamoto, and M. Morisue, "Bit-stream signal processing circuits and their application," *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.*, vol. 85, no. 4, pp. 853–860, 2002.
- [6] H. Fujisaka, N. Masuda, M. Sakamoto, and M. Morisue, "Arithmetic circuits for single-bit digital signal processing," in *Proc. 6th IEEE Int. Conf. Electron. Circuits Syst.*, 1999, vol. 3, pp. 1389–1392.
- [7] P. O'Leary and F. Maloberti, "Bit stream adder for oversampling coded data," *Electron. Lett.*, vol. 26, no. 20, pp. 1708–1709, 1990.
- [8] C. W. Ng, N. Wong, and T. S. Ng, "Efficient FPGA implementation of bit-stream multipliers," *Electron. Lett.*, vol. 43, no. 9, pp. 496–497, 2007.
- [9] D. A. Johns and D. M. Lewis, "Design and analysis of delta-sigma based IIR filters," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 40, no. 4, pp. 233–240, Apr. 1993.
- [10] J. Paulos, G. Brauns, M. B. Steer, and S. Ardalán, "Improved signal-to-noise ratio using tri-level delta-sigma modulation," in *Proc. IEEE Int. Symp. Circuits Syst.*, 1987, pp. 463–466.
- [11] P. W. Wong, "Fully sigma-delta modulation encoded FIR filters," *IEEE Trans. Signal Process.*, vol. 40, no. 6, pp. 1605–1610, Jun. 1992.
- [12] S. R. Norsworthy, R. Schreier, and G. C. Temes, *Delta-Sigma Data Converters, Theory, Design, and Simulation*. Piscataway, NJ: IEEE Press, 1997.
- [13] *Virtex-5 Device Information*, [Online]. Available: <http://www.xilinx.com>
- [14] F. M. Gardner, *Phaselock Techniques*. Hoboken, NJ: Wiley-Interscience, 2005.
- [15] C. Dick and F. Harris, "Direct digital synthesis — some options for FPGA implementation," *Proc. SPIE Int. Symp. Voice Video and Data Commun.*, vol. 3844, pp. 2–10, 1999.
- [16] J. Gonçalves, J. R. Fernandes, and M. M. Silva, "A reconfigurable quadrature oscillator based on a direct digital synthesis system," in *Proc. Conf. Design Circuits Integr. Syst.*, 2006 [Online]. Available: <http://www.inesc-id.pt/ficheiros/publicacoes/3627.pdf>